Fast and accurate coarsening simulation with an unconditionally stable time step

Benjamin P. Vollmayr-Lee*

Department of Physics, Bucknell University, Lewisburg, Pennsylvania 17837, USA

Andrew D. Rutenberg†

Department of Physics and Atmospheric Science, Dalhousie University, Halifax, Nova Scotia, Canada B3H 3J5

We present Cahn-Hilliard and Allen-Cahn numerical integration algorithms that are unconditionally stable and so provide significantly faster accuracy-controlled simulation. Our stability analysis is based on Eyre's theorem and unconditional von Neumann stability analysis, both of which we present. Numerical tests confirm the accuracy of the von Neumann approach, which is straightforward and should be widely applicable in phase-field modeling. For the Cahn-Hilliard case, we show that accuracy can be controlled with an unbounded time step Δt that grows with time t as $\Delta t \sim t^\alpha$. We develop a classification scheme for the step exponent α and demonstrate that a class of simple linear algorithms gives $\alpha = 1/3$. For this class the speedup relative to a fixed time step grows with N, the linear size of the system, as $N/\ln N$. With conservative choices for the parameters controlling accuracy and finite-size effects we find that an 8192^2 lattice can be integrated 300 times faster than with the Euler method.

I. INTRODUCTION

A starting point in the analysis of coarsening systems, such as the phase-separation dynamics following a quench from a disordered to an ordered phase, is the characterization of the asymptotic *late-time* behavior. Most coarsening systems exhibit asymptotic dynamical scaling with the characteristic length scale $L(t)$ given by the size of individual ordered domains. The growth law $L \sim t^n$ is determined by only a few general features, such as conservation laws and the nature of the order parameter (see Ref. [1] for a review). For conserved Cahn-Hilliard equations describing phase separation, $L \sim t^{1/3}$ at late times. More detailed information about the scaling state is difficult to obtain analytically. Indeed, the very existence of scaling has only been demonstrated empirically in simulations and experiments. Consequently, computer simulations of coarsening models, especially phase-field models such as the Cahn-Hilliard equation, play an essential role in our understanding and characterization of late-stage coarsening.

These simulations face several restrictions. To accurately resolve the asymptotic structure it is necessary to evolve until late times so that $L(t) \gg w$, where w is the domain wall width. However, to avoid finite-size effects we must halt the simulation when $L(t)$ is some fraction of the system size L_{sys}. Additionally, to resolve the domain wall adequately the lattice spacing Δx must be sufficiently small compared to the domain wall width w. Very large lattices of linear size $L_{sys}/\Delta x$ are necessary to satisfy all of these requirements: $\Delta x < w \ll L(t) < L_{sys}$. Accurate studies of the scaling state require us to evolve large systems to late times.

Unfortunately, current computational algorithms are very inefficient in their time integration. The standard Euler integration of the Cahn-Hilliard (CH) and Allen-Cahn (AC) coarsening models for conserved and nonconserved dynamics, respectively, is known to be unstable for time steps Δt above a threshold fixed by the lattice spacing Δx—this is the "checkerboard" instability [2]. This imposes a fixed time step irrespective of the natural time scale set by the physical dynamics. The domain walls move increasingly slowly, for example, the CH equation yields asymptotic domain wall velocities $v \sim \partial L/\partial t \sim t^{-2/3}$. Consequently, a fixed time step results in ever-decreasing amounts of domain wall motion per step and eventually becomes wastefully accurate.

Ideally, one would like a stable integration algorithm, which would allow *accuracy* requirements rather than *stability* limitations to determine the integration step size. Recently, Eyre proved the existence of unconditionally gradient stable algorithms (essentially a strict nonincrease in free energy for every possible time step) [3], and provided explicit examples of stable steps for both CH and AC dynamics [3,4]. The present work is concerned with developing these methods in two directions: clarifying and expanding the class of unconditionally stable algorithms and deriving the accuracy limitations on these algorithms.

Our main results for stability are the following. We have determined the parameter range for which Eyre's theorem proves unconditional gradient stability (Sec. II A), and we present Eyre's theorem in Appendix A. We have also determined the parameter range that is unconditionally von Neumann (vN) stable, that is, linearly stable for any size time step (Sec. II B). The latter range is a superset of the former, and neither appear to have been previously determined. We have also performed numerical tests of stability in dimension $d = 2$ (Sec. II C) and found that the vN stability condition appears to be sufficient for identifying unconditionally gradient stable steps. Specifically, for the parameterless form of the CH equation (see Ref. [1])

---

*Electronic address: bvollmay@bucknell.edu

†Electronic address: andrew.rutenberg@dal.ca; URL: http://www.physics.dal.ca/~adr
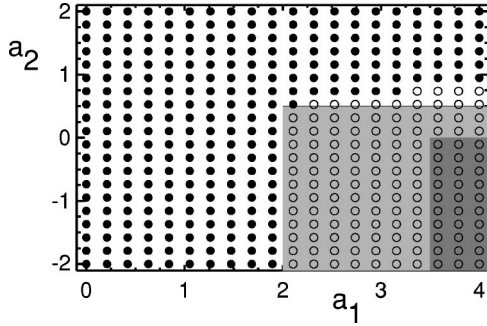
FIG. 1. For time steps parametrized as in Eq. (2) the dark shaded region indicates parameters for which Eyre's theorem proves unconditional gradient stability, while the light shaded region corresponds to unconditional von Neumann (linearly) stable steps. The open circles denote steps that are numerically gradient stable under all of our tests, as described in Sec. II C, while the black circles indicate parameters that were found numerically not to be gradient stable.

$$\partial_t \phi = -\nabla^2(\nabla^2\phi + \phi - \phi^3), \qquad (1)$$

there exists a class of semi-implicit steps

$$\tilde{\phi}_{t+\Delta t} + \Delta t \nabla^2[(1-a_1)\tilde{\phi}_{t+\Delta t} + (1-a_2)\nabla^2\tilde{\phi}_{t+\Delta t}]$$
$$= \phi_t + \Delta t \nabla^2[-a_1\phi_t - a_2\nabla^2\phi_t + \phi_t^3] \qquad (2)$$

that may be solved for the updated field $\tilde{\phi}_{t+\Delta t}$ efficiently by means of fast Fourier transform (FFT). The various stability conditions for these steps are depicted in terms of a_1 and a_2 in Fig. 1. The stability conditions do not depend on the lattice type or dimension, on the volume fraction, or on the form of the lattice Laplacian. This implies, for example, that these algorithms could be combined with adaptive mesh techniques (see, for example, Ref. [5]) for independent control of spatial and temporal discretization. Figure 1 suggests that the unconditional vN stability conditions, which are widely applicable and relatively easy to analyze, may provide a reasonably accurate proxy for unconditional gradient stability. We have also determined the analogous stability conditions for the AC equation.

When stability is not the limiting factor, practical limits are still imposed by *accuracy*. To maintain the domain wall profile to a given accuracy, a time step should be chosen so that the wall only moves a fraction of its width w in a single step. For a scaling system with $L \sim t^n$, where $n \leq 1$ generally, the passage time τ scales like $w/v \sim w/\dot{L} \sim t^{1-n}$ at late times. Then the natural time step should scale as

$$\Delta t_{nat} \sim \tau \sim t^{1-n}. \qquad (3)$$

For CH dynamics $n = 1/3$ and $\Delta t_{nat} \sim t^{2/3}$, while for AC dynamics $n = 1/2$ and $\Delta t_{nat} \sim t^{1/2}$. However, we show that these stable algorithms are still not capable of accurately simulating coarsening using the natural time scale—despite their stability. For example, accuracy limits the stable CH steps given above to "only" $\Delta t \sim t^{1/3}$.

To understand the limitations imposed on even stable algorithms by accuracy, we study in Sec. III the truncation error for the CH equation for general numerical algorithms, and determine how these terms scale with time to all orders in Δt (Sec. III D). We develop a classification scheme for such algorithms based on the lowest order p of Δt^p at which truncation error fails to follow its optimal scaling, and show that this term limits the accuracy of the algorithm at late times (Sec. III A). Our analysis leads to the conclusion that accuracy requires a time step

$$\Delta t \sim t^{2(p-1)/3p} \qquad (4)$$

for the CH model. The algorithms in Eq. (2) have $p = 2$, meaning the error becomes suboptimal at $O(\Delta t^2)$, the leading error term. This result is consistent with our numerical observations. Our simple analysis for the natural time step, Eq. (3), corresponds to the $p = \infty$ class. We are unable to identify any such "perfect" algorithms for the CH case; they may be impossible for any nonlinear problem.

Next, we turn to the question of practical advantage. Various computational algorithms have been developed to mitigate the impact of instabilities by increasing Δt by a fixed factor compared to the simplest Euler discretization. For example, the cell-dynamical scheme (CDS) [6] exploits universality to choose a free energy that is convenient in terms of numerical stability. More recently, Fourier spectral methods [7,8] have been shown to increase the maximum Δt by an impressive two orders of magnitude. However, these methods still require fixed time steps and so cannot adjust to the naturally slowing CH dynamics.

In Sec. IV we determine the relative advantage of integration by algorithms such as Eq. (2) compared to the conventional Euler method. For a reasonably conservative choice of accuracy requirements, we find for an 8192×8192 lattice (currently feasible for a Linux workstation) with $\Delta x = 1$ that our unconditionally stable methods can integrate up to finite-size effects roughly a factor of 300 times faster than possible with the Euler method. The advantage of unconditionally stable steps increases with larger system sizes: for lattices of linear size N we show the relative advantage in speed is of order $N/\ln N$, regardless of the spatial dimension of the system. This means that as computational power continues to increase, unconditionally gradient stable algorithms will become even more valuable.

We present a summary and outlook for future developments and applications in Sec. V.

II. STABILITY

The parameterless form of the CH equation for a conserved scalar field [1] is

$$\partial_t \phi = \nabla^2 \mu, \qquad (5)$$

where μ is the local chemical potential given by

$$\mu(\mathbf{x}) \equiv \frac{\delta F}{\delta \phi(\mathbf{x})} \qquad (6)$$

and $F[\phi]$ is the free-energy functional, taken here to be

$$F[\phi] \equiv \int d^d x [\tfrac{1}{2}(\boldsymbol{\nabla}\phi)^2 + \tfrac{1}{4}(\phi^2 - 1)^2]. \qquad (7)$$

The second term in F represents a double-well potential with equilibrium values $\phi = \pm 1$, and Eqs. (5)–(7) combine to give Eq. (1). The parameterless form of the AC equation [1] is

$$\partial_t \phi = -\mu = \nabla^2 \phi + \phi - \phi^3. \qquad (8)$$

For dissipative dynamics such as the CH and AC equations, a discrete time stepping algorithm is defined to be *gradient stable* only if the free energy is nonincreasing, $F[\tilde{\phi}_{t+\Delta t}] \leq F[\phi_t]$, for any field configuration ϕ_t. The other requirements for gradient stability, e.g., that stable fixed points must correspond to minima of F or that F should increase without bound for large ϕ, are already manifest in the discretized forms of these equations. Gradient stability may reasonably be regarded as the ultimate stability criterion for the CH equation.

Unconditional gradient stability means that the conditions for gradient stability hold for *any* size time step $\Delta t \in [0,\infty)$. Since unconditionally stable steps are our primary concern, we will henceforth use "stable" or "unstable" to refer to the behavior for arbitrarily large Δt. That is, stable implies unconditionally stable, while a fixed time-step algorithm such as the Euler step may be referred to as unstable or conditionally stable.

The Euler time discretization of the CH equation is

$$\phi_{t+\Delta t}^{\mathrm{Eu}} \equiv \phi_t + \Delta t \nabla^2 \mu_t. \qquad (9)$$

The Euler update is "explicit" since the field at the earlier time step (ϕ_t) explicitly determines the field at the next time step ($\phi_{t+\Delta t}$). It is also unstable for values of Δt that exceed a lattice-dependent threshold, $\Delta t_{max} \sim \Delta x^4$ [2]. The fully implicit time step is obtained by replacing μ_t with $\mu_{t+\Delta t}$ in Eq. (9), and is, like the Euler step, accurate to $O(\Delta t)$. Other time steps, which involve splitting μ into parts evaluated at t and at $t+\Delta t$, are generally called semi-implicit methods.

Remarkably, Eyre [3,4] proved that appropriate semi-implicit parametrizations can lead to stable update steps for both CH and AC equations. To explore these possibilities, it is useful to introduce a general family of such steps for the CH equation in an arbitrary spatial dimension:

$$\tilde{\phi}_{t+\Delta t} + (1-a_1)\Delta t \nabla^2 \tilde{\phi}_{t+\Delta t} + (1-a_2)\Delta t \nabla^4 \tilde{\phi}_{t+\Delta t}$$
$$- (1-a_3)\Delta t \nabla^2 [\phi_t^m \tilde{\phi}_{t+\Delta t}^{3-m}]$$
$$= \phi_t - a_1\Delta t \nabla^2 \phi_t - a_2\Delta t \nabla^4 \phi_t + a_3\Delta t \nabla^2 \phi_t^3. \qquad (10)$$

This reduces to Eq. (2) for $a_3 = 1$. For each of the three terms on the right-hand side of Eq. (1) there generally are *both* explicit and implicit contributions to Eq. (10), and this will be exploited to construct stable dynamics for any size Δt. For all values of the parameters a_i and m this step gives a solution $\tilde{\phi}_{t+\Delta t}$ that is $O(\Delta t)$ accurate. The implicit terms

are denoted by $\tilde{\phi}_{t+\Delta t}$, with $\phi_{t+\Delta t}$ reserved to represent the exact field obtained by integration of Eq. (1) over the time step Δt. We choose our parametrization such that $a_1 = a_2 = a_3 = 1$ corresponds to the Euler update, Eq. (9), while $a_1 = a_2 = a_3 = 0$ is the fully implicit step. For $a_3 \neq 1$ we have, motivated by Eyre, a mixed nonlinear term with $0 \leq m < 3$ that combines implicit and explicit terms.

It is useful to sort algorithms described by Eq. (10) into three categories based on how they are implemented numerically. First, when $a_3 = 1$ we have *linear direct* steps, where the equation for $\tilde{\phi}_{t+\Delta t}$ is linear and has spatially uniform coefficients so the updated field can be found efficiently, and with no additional memory requirements, by FFT methods [9,10]. Second, when $a_3 \neq 1$ but $m = 2$ then the implicit equation remains linear in $\tilde{\phi}_{t+\Delta t}$ but no longer has spatially uniform coefficients. Eyre outlines an iterative procedure for solving these equations [4], so we call these *linear iterative* steps. Insisting on convergence of the iterative procedure restricts this class to a subset of parameter values. Finally, for $a_3 \neq 1$ and $m \neq 2$ the update equation is *nonlinear*. For some parameter values the nonlinear equation can lead, unphysically, to multiple solutions. This occurs for both the fully implicit case $a_1 = a_2 = a_3 = m = 0$, and the Crank-Nicholson case $a_1 = a_2 = a_3 = 1/2$, $m = 0$, whenever Δt exceeds a threshold value [3]. Generally, the nonlinear equations require solution by the Newton-Raphson method [9,10], which is complicated to implement in two or more spatial dimensions. For some parameter values this can be demonstrated to be absolutely convergent (as shown below), so nonlinear steps provide a viable option—though not one we have explored numerically.

The step parametrization for the AC equation analogous to Eq. (10) is

$$\tilde{\phi}_{t+\Delta t} - (1-a_1)\Delta t \tilde{\phi}_{t+\Delta t} - (1-a_2)\Delta t \nabla^2 \tilde{\phi}_{t+\Delta t}$$
$$+ (1-a_3)\Delta t [\phi_t^m \tilde{\phi}_{t+\Delta t}^{3-m}]$$
$$= \phi_t + a_1\Delta t \phi_t + a_2\Delta t \nabla^2 \phi_t - a_3\Delta t \phi_t^3, \qquad (11)$$

which we include because the theoretical stability analysis follows nearly identically for the CH and AC equations, and the stability regions are given by the same shaded regions of Fig. 1.

A. Unconditionally stable steps from Eyre's theorem

Eyre's theorem (see Appendix A) shows that an unconditionally gradient stable algorithm results, for both CH and AC equations, if one can split the free energy appropriately into *contractive* and *expansive* parts, $F = F^C + F^E$, and treat the contractive parts implicitly and the expansive parts explicitly. That is, the CH equation (5) is discretized as

$$\tilde{\phi}_{t+\Delta t} - \Delta t \nabla^2 \mu_{t+\Delta t}^C = \phi_t + \Delta t \nabla^2 \mu_t^E, \qquad (12)$$

while the AC equation (8) is discretized as

$$\tilde{\phi}_{t+\Delta t} + \Delta t \mu_{t+\Delta t}^C = \phi_t - \Delta t \mu_t^E, \qquad (13)$$

where $\mu_i^X = \partial F^X / \partial \phi_i$ for lattice site i, and where ∇^2 implies a lattice Laplacian. The necessary condition on the splitting is the same for both equations and may be stated by introducing the Hessian matrices

$$M_{ij} = \frac{\partial^2 F}{\partial \phi_i \partial \phi_j}, \quad M_{ij}^E = \frac{\partial^2 F^E}{\partial \phi_i \partial \phi_j}, \quad M_{ij}^C = \frac{\partial^2 F^C}{\partial \phi_i \partial \phi_j}, \tag{14}$$

where i,j denote lattice sites. First, we must have all eigenvalues of $\mathbf{M}^E$ nonpositive and all eigenvalues of $\mathbf{M}^C$ nonnegative. Second, as shown in Appendix A, for λ_{min} equal to the smallest eigenvalue of $\mathbf{M}$ and λ_{max}^E the largest eigenvalue of $\mathbf{M}^E$, we need

$$\lambda_{max}^E \leq \frac{1}{2} \lambda_{min}. \tag{15}$$

This also automatically satisfies the convexity requirement for $\mathbf{M}^E$, since $\lambda_{min} < 0$.

To identify the appropriate splittings, it is useful to break the free energy, Eq. (7), in its lattice-discretized form, into three parts (neglecting the irrelevant constant $V/4$ term):

$$F^{(1)} = -\sum_i \frac{1}{2} \phi_i^2, \quad F^{(2)} = \sum_i \frac{1}{2} (\nabla \phi)_i^2,$$

$$F^{(3)} = \sum_i \frac{1}{4} \phi_i^4, \tag{16}$$

with corresponding Hessian matrices $\mathbf{M}^{(i)}$. First, $M_{ij}^{(1)} = -\delta_{ij}$, where δ_{ij} is the Kronecker δ function, has all eigenvalues equal to -1. Next, $M_{ij}^{(2)} = (-\nabla^2)_{ij}$ is the negative of the lattice Laplacian, which can always be diagonalized by going to Fourier space. It immediately follows that the eigenvalues of $\mathbf{M}^{(2)}$ are strictly non-negative. (Even for irregular spatial discretizations, the $\mathbf{M}^{(2)}$ eigenvalues must be non-negative.) Finally, $M_{ij}^{(3)} = 3\phi_i^2 \delta_{ij}$, which has strictly non-negative eigenvalues as well. We parametrize the splitting via

$$F^E = \sum_{i=1}^3 a_i F^{(i)}, \quad F^C = \sum_{i=1}^3 (1 - a_i) F^{(i)} \tag{17}$$

which results in the general CH step, Eq. (10), and AC step, Eq. (11), when $m = 0$.

Now to obtain bound, since the sum of matrices, $\mathbf{M} = \mathbf{M}^{(1)} + \mathbf{M}^{(2)} + \mathbf{M}^{(3)}$, has eigenvalues bounded by the sum of the bounds, the minimum eigenvalue of $\mathbf{M}$ satisfies $\lambda_{min} \geq -1$. Therefore Eq. (15) is satisfied by ensuring $\lambda_{max}^E \leq -1/2$.

One example that satisfies these conditions is the splitting $F^E = F^{(1)}$ and $F^C = F^{(2)} + F^{(3)}$, since $\lambda_{max}^E = -1$ satisfies Eq. (15) and $\mathbf{M}^C$ has strictly non-negative eigenvalues. This provides a gradient stable nonlinear step with $a_1 = 1$ and $a_2 = a_3 = 0$. This case was identified by Eyre [3], who noted that the convexity requirement for the splitting guarantees absolute convergence of the Newton-Raphson method.

Eyre also presents a technique for identifying stable linear *direct* algorithms [3], which relies on the fact that ϕ^2 is bounded. It exceeds unity only slightly in the CH equation and only in the interior region of a curved interface due to Gibbs-Thompson effects [11]. Therefore the eigenvalues of $\mathbf{M}^{(3)}$ have an effective upper bound, approximately three. If we then take $F^E = a_1 F^{(1)} + F^{(3)}$ (so $a_3 = 1$ and $a_2 = 0$) the eigenvalues of $\mathbf{M}^E$ are of the form $-a_1 + 3\phi_i^2$ and satisfy Eq. (15) for $\phi_i^2 \lesssim 1$ if $a_1 \gtrsim 7/2$. Any value $a_2 \leq 0$ will give the same result, since negative values of a_2 can only decrease the eigenvalues of $\mathbf{M}^E$. These choices imply $F^C = (1 - a_1)F^{(1)} + (1 - a_2)F^{(2)}$, which has the necessary nonnegative eigenvalues for the range of a_1 and a_2 given above. Therefore we can identify a class of gradient stable direct CH and AC steps as

$$a_1 \gtrsim 7/2, \quad a_2 \leq 0, \quad a_3 = 1. \tag{18}$$

This gives the dark gray shaded region in Fig. 1. These represent *sufficient* restrictions on the a_i to satisfy the conditions for Eyre's theorem; however other values of the a_i may be gradient stable as well.

Eyre provided specific step examples for all three implementation categories: a nonlinear step, with $a_1 = 1$, $a_2 = a_3 = 0$, and $m = 0$, a linear *iterative* step with $m = 2$ and the same a_i as the nonlinear step, and a linear *direct* step with $a_1 = 3$, $a_2 = 0$, and $a_3 = 1$ [3,4]. The nonlinear step is the example presented earlier in this section, and its gradient stability follows from Eyre's theorem. However, it is not clear to us that Eyre's theorem can be directly applied to the iterative steps, and in fact we find Eyre's proposed iterative method to be numerically unstable, as described in Sec. II C. Finally, the a_1 value in the direct step violates Eq. (18), so this case does not follow from Eyre's theorem.

B. Unconditional von Neumann stability

Since Eyre's theorem provides, in principle, only a subset of the possible gradient stable steps, complementary approaches for determining stability are desirable. In this section we extend vN linear stability analysis [9,10] to arbitrary time steps, which we call unconditional vN stability. Since any gradient stable algorithm is likely also linearly stable, the von Neumann analysis would appear to identify a *superset* of possibly gradient stable algorithms: in principle, the vN analysis could also identify some unwanted nonlinearly unstable algorithms. As shown in Fig. 1, though, the vN stability boundary corresponds quite well with the numerically determined gradient stability line. This leads us to suggest that the approach of imposing unconditional vN stability on a broadly parametrized class of semi-implicit algorithms, followed by numerical checking, could be fruitfully adapted to a wide variety of applications.

We analyze the general step, Eq. (10), for linear stability around a constant phase $\phi = \phi_0$. It is important to realize there are *physical*, and therefore desirable, linear instabilities in the continuum CH and AC equations. Therefore it is important to distinguish between these and the unphysical instabilities induced by the numerical implementation. Take

$\phi(\mathbf{x},t) \equiv \phi_0 + \eta(\mathbf{x},t)$, and linearize the CH equation (1) in η to get $\dot{\eta} = -\nabla^2(\nabla^2\eta + \eta - 3\phi_0^2\eta)$. Fourier transform this to get

$$\dot{\eta}_{\mathbf{k}} = -\lambda_{\mathbf{k}}(\lambda_{\mathbf{k}} + k_0^2)\eta_{\mathbf{k}}, \tag{19}$$

$$k_0^2 \equiv 1 - 3\phi_0^2. \tag{20}$$

Here $\lambda_{\mathbf{k}}$ is the eigenvalue of the Laplacian and is nonpositive, with $\lambda_{kmin} \leq \lambda_{\mathbf{k}} \leq 0$ (note that $\lambda_{\mathbf{k}} = -k^2$ in the continuum). The minimum value λ_{kmin} depends on the lattice, spatial dimension, and specific form of the Laplacian. Similarly, for the same ϕ linearize the AC equation (8) in η and Fourier transform to get

$$\dot{\eta}_{\mathbf{k}} = (\lambda_{\mathbf{k}} + k_0^2)\eta_{\mathbf{k}}. \tag{21}$$

The physical instability for both Eqs. (19) and (21) occurs for

$$-\lambda_{\mathbf{k}} < k_0^2, \tag{22}$$

which corresponds in the CH equation to spinodal decomposition [1]. We stress that while these Fourier modes are linearly unstable, the dynamics of spinodal decomposition is *gradient stable* and represents a physical decrease of the free energy, which is why it must be retained.

We now linearize and Fourier transform our general CH step, Eq. (10), as above to get

$$[1 - \lambda_{\mathbf{k}}\Delta t\{(a_1 - 1) - \lambda_{\mathbf{k}}(1 - a_2)$$
$$+ \phi_0^2(1 - a_3)(3 - m)\}]\eta_{\mathbf{k},t+\Delta t}$$
$$= [1 - \lambda_{\mathbf{k}}\Delta t\{a_1 + \lambda_{\mathbf{k}}a_2 + \phi_0^2$$
$$\times [-3a_3 + m(a_3 - 1)]\}]\eta_{\mathbf{k},t}. \tag{23}$$

Writing this as

$$[1 + \Delta t\mathcal{L}_{\mathbf{k}}]\eta_{\mathbf{k},t+\Delta t} = [1 + \Delta t\mathcal{R}_{\mathbf{k}}]\eta_{\mathbf{k},t}, \tag{24}$$

the von Neumann stability criterion is

$$|1 + \Delta t\mathcal{L}_{\mathbf{k}}| > |1 + \Delta t\mathcal{R}_{\mathbf{k}}|, \tag{25}$$

which implies that small deviations from the constant solution evolve to decrease in magnitude. We want to impose this stability condition for all $\mathbf{k}$ and *arbitrary* positive Δt. For large Δt, Eq. (25) implies $|\mathcal{L}_{\mathbf{k}}| > |\mathcal{R}_{\mathbf{k}}|$. The left-hand side of Eq. (25) can be made to violate the inequality for small Δt unless $\mathcal{L}_{\mathbf{k}} \geq 0$. Combining these conditions we have

$$\mathcal{L}_{\mathbf{k}} > |\mathcal{R}_{\mathbf{k}}|, \tag{26}$$

which is a *necessary and sufficient* condition for unconditional linear stability. This condition applies to *all* first-order time steps that can be expressed in the form given by Eq. (24).

We examine the linear stability condition in two steps. First, $\mathcal{L}_{\mathbf{k}} > \mathcal{R}_{\mathbf{k}}$:

$$0 < \mathcal{L}_{\mathbf{k}} - \mathcal{R}_{\mathbf{k}} = (-\lambda_{\mathbf{k}})[-1 - \lambda_{\mathbf{k}} + 3\phi_0^2]. \tag{27}$$

This reduces to the spinodal condition, Eq. (22). Note that all the parameters (a_1, a_2, a_3, m) are absent from Eq. (27), so we cannot interfere with the spinodal condition. This evidently follows from having a first-order accurate step. Next, we check for $\mathcal{L}_{\mathbf{k}} > -\mathcal{R}_{\mathbf{k}}$, which gives

$$2a_1 - 1 - [(3 - m)(2a_3 - 1) + m]\phi_0^2 + \lambda_{\mathbf{k}}(2a_2 - 1) > 0. \tag{28}$$

If we choose $a_2 < 1/2$, then since $\lambda_{\mathbf{k}} \leq 0$ we get $2a_1 - 1 - [(3 - m)(2a_3 - 1) + m]\phi_0^2 > 0$. For $a_2 > 1/2$ we obtain a lattice-dependent condition, that is, our inequality would contain λ_{kmin}.

We choose to restrict ourselves to lattice-*independent* stability conditions as these are more practical: they carry over into any lattice or spatial dimension. For this purpose we take $a_2 < 1/2$. This gives the vN stable conditions

$$a_2 < 1/2,$$

$$a_1 > \frac{1 + \max[0, (3 - m)(2a_3 - 1) + m]}{2}. \tag{29}$$

We have let ϕ_0^2 vary in the late-time asymptotic range of $\phi_0^2 \in [0,1]$, where Gibbs-Thompson-induced supersaturation has been ignored, and have imposed on a_1 the most restrictive value that results. For this reason algorithms near the stability boundaries should be avoided at early times.

For direct steps, with $a_3 = 1$, the second condition in Eq. (29) becomes $a_1 > 2$. This gives the lightly shaded region in Fig. 1. The Euler update, with $a_1 = a_2 = a_3 = 1$, is clearly unstable since $a_2 > 1/2$ and $a_1 < 2$. For linear iterative steps, with $m = 2$, Eq. (29) becomes $a_1 > \max[1/2, a_3 + 1]$. The stability condition of the general nonlinear step cannot be further simplified from Eq. (29), but the special case $m = 0$ gives $a_1 > \max[1/2, 3a_3 - 1]$.

There is another special case for which the stability conditions can be imposed, namely, when $m = 0$ and $a_1 = a_2 = a_3 \equiv a$. In this case the vN condition, Eq. (28), becomes

$$(1 - 2a)[-1 + 3\phi_0^2 + \lambda_{\mathbf{k}}] > 0. \tag{30}$$

The term in square brackets is again the spinodal condition and should be positive for all physically stable modes, so for $a < 1/2$ both vN stability conditions reduce to the spinodal condition. However, these steps, which include the marginal Crank-Nicholson case ($a = 1/2$) and the stable fully implicit step ($a = 0$) suffer from having multiple solutions to the nonlinear implicit equation whenever Δt exceeds some threshold, making them unsuitable.

Regarding Eyre's proposed steps, introduced at the end of Sec. II A, we note that the direct step is vN stable, the iterative step is marginal for vN stability, and the nonlinear step, which was gradient stable by Eyre's theorem, is also vN stable.

The linearization for the general AC step, Eq. (11), results in the same linearized equation (23) but with the substitution

$-\lambda_{\mathbf{k}}\Delta t \to \Delta t$. Since $-\lambda_{\mathbf{k}} \geq 0$, the vN stability analysis for the AC equation is identical and also results in Eq. (29).

C. Numerical stability tests

The vN stability analysis yields a considerably larger parameter range for stable steps, Eq. (29), than those which are provably stable by Eyre's theorem, e.g., Eq. (18). Here we determine numerically which step parametrizations are gradient stable, for purposes of comparison with the theoretical results. We focus primarily on direct steps, with $a_3 = 1$, since these are an important practical class of steps. We consider only symmetric quenches of the CH equation in this section, with $\langle \phi \rangle = 0$.

The primary result, shown in Fig. 1, is obtained as follows. We evolved a uniformly distributed 20×20 array of direct CH steps with the parameter values $a_1 \in (0,4)$ and $a_2 \in (-2,2)$ on a 512^2 lattice to a final time t_{max}. We take lattice spacing $\Delta x = 1$ here and throughout. At regular intervals during the evolution we tested a single direct step with $0 < \Delta t < 10^{10}$. This step was only used for stability testing, and did not contribute to the time evolution. Steps larger than $\Delta t = 10^{10}$ were not employed to avoid spurious roundoff error effects. Any system that *ever* increased its free energy was labeled unstable, and plotted in Fig. 1 with a filled circle. The systems were evolved in time with multiple methods. First, we used Euler updates ($\Delta t = 0.05$) evolved to $t_{max} = 10^4$. Next, we evolved systems with direct updates both with fixed $\Delta t = 100$ and with an increasing time step $\Delta t = 0.05 \, t^{1/3}$ (both to $t_{max} = 10^6$).

As Fig. 1 shows, all vN stable algorithms were found numerically to be gradient stable, and the lightly shaded region corresponds extremely well to the gradient stable systems. Indeed, the vN stability boundary for a_1 appears to be followed quite sharply in the numerical tests. We do find numerical gradient stability for a region where $a_2 > 1/2$: this is most likely due, ironically, to a lattice-induced stabilization. That is, since the lattice Laplacian $\lambda_{\mathbf{k}}$ has an implementation-dependent minimum value, the inequality (28) may be satisfied for some $a_2 > 1/2$. Therefore, we expect the precise location of this boundary to shift slightly depending on the lattice, the spatial dimension, and the choice of lattice Laplacian, but not to cross $a_2 = 1/2$.

With the numerical tests described above we have tested the linear iterative step proposed by Eyre [4] and found it to be unstable.

To help illustrate numerical testing of gradient stability, we show a mixture of stable and unstable steps in Figs. 2 and 3. The difference between gradient stable and unstable steps is striking: while neither are particularly accurate for extremely large Δt, the unstable steps show a marked increase in the free-energy density, while the gradient stable steps adhere to the strict nonincreasing free-energy condition. However, the closer view in Fig. 3 shows that some cost is paid in accuracy: for small values of Δt, both the Euler step and the unstable semi-implicit step track the physical behavior better than the stable step. While it may appear from Fig. 3 that moderately large steps may be used with unstable algorithms, this is not case; for example, using a $\Delta t \gtrsim 0.05$ for
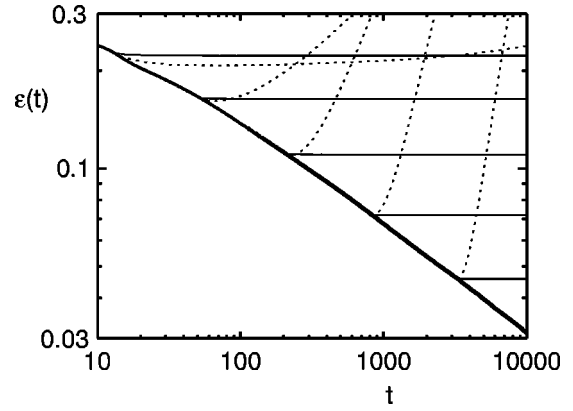

FIG. 2. Plot of the free-energy density ϵ vs time (thick solid line) approaching the asymptotic $\epsilon \sim t^{-1/3}$ decay, as evolved with a Euler update with $\Delta t = 0.01$ in a 1024^2 system. At five distinct departure times t_d, separated by factors of 4, we show the free energies that result from a *single* time step $\Delta t \in (0, 10\,000)$, plotted vs $t = t_d + \Delta t$. The dotted lines correspond to using a common semi-implicit algorithm ($a_1 = 1$, $a_2 = 0$, $a_3 = 1$) for the single step, while the thin solid lines correspond to single steps with a vN stable direct algorithm ($a_1 = 3$, $a_2 = 0$, and $a_3 = 1$).

the Euler update will lead rapidly to instability via accumulated error from *repeated* steps, due to the checkerboard instability. In contrast, by nature of its vN stability, the stable step does not have this rapid accumulation of error. Clearly, though, the extreme step sizes used to illustrate stability in Figs. 2 and 3 would not be appropriate for an accurate simulation. In the following section we show how accuracy provides a time-dependent (but unbounded) limit on the step size.

III. ACCURACY

With a gradient stable algorithm, it is possible to use a progressively larger time step as the characteristic dynamics become slower. The limiting factor for the increase of the time step is then an *accuracy* requirement.
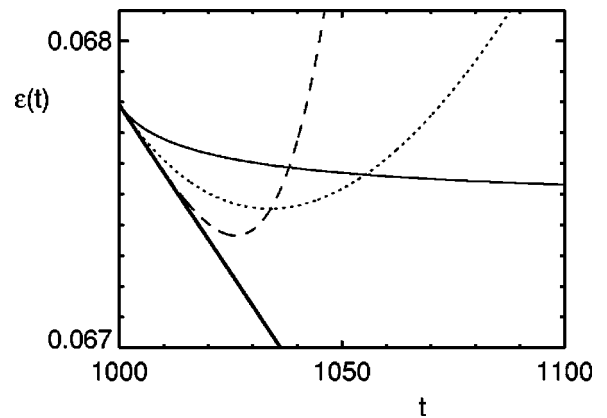

FIG. 3. As per Fig. 2, but with $t_d = 1000$. The dashed line corresponds to a single step of the Euler update, which is gradient unstable. Both the Euler step and the unstable semi-implicit step (dotted) are unstable under repeated steps for much smaller Δt than appear to be accurate for a single step.

A specified accuracy criterion may be imposed on the stable steps identified in Sec. II without any further theoretical development using standard numerical adaptive step-size techniques (as described in Ref. [9] and discussed in Sec. III B). For the CH equation, one would naively expect a time step growing as $\Delta t \sim t^{2/3}$, for the reasons presented in Sec. I. However, this is not the case: empirically we find significantly slower growth. This motivated us to study the sources of error terms in the gradient stable CH steps. Our main result is the p classification scheme, which determines the allowed growth rate of the time step according to Eq. (4). The remainder of this section is concerned with the CH equation.

A. The p classification scheme

We begin with an analysis of the error magnitude associated with the various gradient stable algorithms. The exact $\phi_{t+\Delta t}$, obtained by integration of Eq. (1) from a given ϕ_t, can be expressed in terms of the fields at time t by means of a Taylor expansion

$$\phi_{t+\Delta t} = \phi_t + \Delta t\, \partial_t \phi_t + \frac{1}{2}\Delta t^2 \partial_t^2 \phi_t + \frac{1}{3!}\Delta t^3\, \partial_t^3 \phi_t + \cdots. \tag{31}$$

The Euler update, Eq. (9), is simply the truncation of this expansion at $O(\Delta t)$ with resulting error $\Delta\phi^{\mathrm{Eu}} \equiv \phi_{t+\Delta t}^{\mathrm{Eu}} - \phi_{t+\Delta t}$ given by

$$\Delta\phi^{\mathrm{Eu}} = -\sum_{n=2}^{\infty} \frac{\Delta t^n}{n!}\partial_t^n \phi_t. \tag{32}$$

Other step parametrizations will have different coefficients for the $O(\Delta t^n)$ component of the error, but the general feature of an expansion to all powers of Δt will be the same. Since our goal is to have a growing time step with controlled error, successively higher powers of Δt will require coefficients decaying increasingly faster in time. In order to determine the limitation on how fast the time step may grow, it is essential to know the decay rates of the coefficients of Δt^n for all n. In this section we demonstrate how this can be done. We make use of the following results for asymptotic decay rates, derived in Sec. III D. In the interfacial region (defined in Sec. III D)

$$\partial_t^n \phi \sim t^{-2n/3}, \quad \partial_t^n(\nabla^2)^k \phi^j \sim t^{-2n/3}, \tag{33}$$

whereas in the bulk, that is, all of the system not near an interface, we find

$$\partial_t^n \phi \sim t^{-(1/3)-(2/3)n}, \quad \partial_t^n(\nabla^2)^k \phi^j \sim t^{-(1/3)-2(n+k)/3}. \tag{34}$$

Consider first the Euler step: all the $O(\Delta t^n)$ coefficients are simply proportional to the time derivative $\partial_t^n \phi$ evaluated at t. If numerical stability were not a problem and we simply increased the time step according to the naive $\Delta t \sim t^{2/3}$, we would find in the interfacial region that every order in the Taylor expansion provides an $O(t^0)$ contribution to the error,

whereas in the bulk region every order provides an $O(t^{-1/3})$ contribution. This would present an accurate solution with a $\Delta t \sim t^{2/3}$ time step, except that, of course, the Euler step is not gradient stable for large time steps.

Now consider the general step, Eq. (10). The error term in this step, $\Delta\tilde{\phi} \equiv \tilde{\phi}_{t+\Delta t} - \phi_{t+\Delta t}$, can be written as

$$\Delta\tilde{\phi} = \Delta\phi^{\mathrm{Eu}} - (1-a_1)\Delta t\nabla^2(\tilde{\phi}_{t+\Delta t} - \phi_t)$$
$$- (1-a_2)\Delta t\nabla^4(\tilde{\phi}_{t+\Delta t} - \phi_t)$$
$$+ (1-a_3)\Delta t\nabla^2[\phi_t^m(\tilde{\phi}_{t+\Delta t}^{3-m} - \phi_t^{3-m})]. \tag{35}$$

This peculiar form with implicit $\tilde{\phi}_{t+\Delta t}$ on the right is useful for the error analysis. By using Eq. (10) iteratively, the implicit terms can be replaced by terms that are higher order in Δt involving the field ϕ_t. For example, we can derive the $O(\Delta t^2)$ part of the error, using $\tilde{\phi}_{t+\Delta t} - \phi_t = \Delta t\, \partial_t \phi_t + O(\Delta t^2)$ and $\tilde{\phi}_{t+\Delta t}^{3-m} - \phi_t^{3-m} = (3-m)\Delta t\, \phi_t^{2-m}\partial_t \phi_t + O(\Delta t^2)$. We find the error in our general step to be

$$\Delta\tilde{\phi} = [-\tfrac{1}{2}\partial_t^2 \phi_t + (a_1-1)\nabla^2\partial_t\phi_t + (a_2-1)\nabla^4\partial_t\phi_t$$
$$+ (1-a_3)(3-m)\nabla^2\phi_t^2\partial_t\phi_t]\Delta t^2 + O(\Delta t^3), \tag{36}$$

where the first term comes from Eq. (32). Now compare the time decay of the various terms. At the interface, the $\partial_t^2 \phi_t$ part decays as $t^{-4/3}$, but the other terms all decay as $t^{-2/3}$. Therefore, for general values of the a_i and m, to keep the $O(\Delta t^2)$ interfacial error fixed, the time step is limited to grow as $\Delta t \sim t^{1/3}$. We see that the Euler case was special because it made all but the first term in the $O(\Delta t^2)$ error vanish. Since every term in Eq. (36) decays faster in the bulk than at the interface, we conclude that the error is *interface limited*, i.e., the accuracy criterion at the interface will determine how fast the time step can grow. This is a generic feature, as we will show below.

There are other ways besides using the Euler step to make the $O(\Delta t^2)$ interfacial error decay as $t^{-4/3}$. If the coefficients satisfy

$$a_1 = a_2 = 1-b, \quad a_3 = 1 - 3b/(3-m) \tag{37}$$

for some b, then the various $\partial_t \phi_t$ terms in Eq. (36) add to give $b\partial_t^2 \phi_t$. In this case,

$$\Delta\tilde{\phi} = -(\tfrac{1}{2}-b)\partial_t^2\phi_t\Delta t^2 + O(\Delta t^3), \tag{38}$$

and so the $O(\Delta t^2)$ coefficient decays as $t^{-4/3}$ at the interface and faster in the bulk. From this example we can construct the p classification scheme.

Consider the truncation error term of order Δt^n. This can be obtained by iterating Eq. (35) and can be expressed as a sum of terms of the form $\partial_t^{n-1}(\nabla^2)^k\phi^j$. If these terms appear in the right proportions, they combine via Eq. (1) to become proportional to $\partial_t^n \phi$, which decays faster by a factor of $1/t^{2/3}$ at the interface. This is exactly what occurs in the $n=2$ case above when Eq. (37) is satisfied.

Now consider some value $p \geq 2$ for which all Δt^n error terms with $n < p$ are proportional to $\Delta t^n \partial_t^n \phi_t$, but at order $m \geq p$ this breaks down into a sum of terms of the type $\Delta t^m \partial_t^{m-1} (\nabla^2)^k \phi_t^j$. In this case the order p term provides the leading asymptotic error. Focusing on interfacial region, the order p term goes as $\Delta t^p t^{-2(p-1)/3}$ according to the second term in Eq. (33). Choosing the time step to hold this term at constant error would require $\Delta t \sim t^\alpha$ with $\alpha = 2(p-1)/(3p)$, as displayed in Eq. (4). Now we show that all higher- and lower-order terms in Δt will decay faster than the Δt^p term for this choice of α. For $n < p$, we have from the first term in Eq. (33) $\Delta t^n t^{-2n/3} \sim t^{n(\alpha - 2/3)} = t^{-2n/3p}$, so the $n < p$ terms give ever-decreasing contributions to the error. For $m > p$ the error terms are of the form $\Delta t^m t^{-2(m-1)/3} \sim t^{-2(m-p)/(3p)}$ which decay as well. Hence the asymptotic interfacial error is given by the $O(\Delta t^p)$ term as advertised, and is order t^0. Note that for this interface limited $\Delta t \sim t^\alpha$ all bulk terms to all orders have decaying error terms, thus establishing interface limited error as a generic feature.

B. Quantifying error for direct steps

Direct steps, with $a_3 = 1$ by definition and $a_1 > 2$, $a_2 < 1/2$ for stability, fail to satisfy Eq. (37), and so all direct steps give $p = 2$ algorithms with $\Delta t \sim t^{1/3}$. This means that the asymptotic error magnitude should be given exactly by

$$|\Delta \tilde{\phi}| = \Delta t^2 |(a_1 - 1)\nabla^2 \partial_t \phi + (a_2 - 1)\nabla^4 \partial_t \phi|, \quad (39)$$

with $\Delta t = At^{1/3}$. This gives a fixed amount of error at the interface, and all higher orders of Δt give decaying contributions. Therefore, the error magnitude is proportional to A^2, and we can use numerical measurements of Eq. (39) to develop the constant of proportionality.

We determine error numerically in the usual way [9]: compare the field $\phi^{(1)}$ obtained from a single step of size Δt to the field $\phi^{(2)}$ obtained from two steps of size $\Delta t/2$. It is straightforward to show that if the true error of the step is $E\Delta t^2 + O(\Delta t^3)$, then $\phi^{(1)} - \phi^{(2)} = (E/2)\Delta t^2 + O(\Delta t^3)$. Since we expect exactly Δt^2 error, we simply take $2(\phi^{(1)} - \phi^{(2)})$ to be the true error.

In the bulk, the error decays as $t^{-2/3}$. The interfacial error is not decaying, but the amount of interface decays as $t^{-1/3}$, which means the error magnitude, Eq. (39), averaged over the entire system will also decay as $t^{-1/3}$, all from the interfacial contribution. To determine the error per lattice site in the interfacial region, it is necessary to divide the averaged error by the fraction of the system in the interfacial region. We do that as follows. The asymptotic free-energy density is given by the product of the surface tension σ and interface density: $\epsilon(t) = \sigma A_{int}(t)/L_{sys}^d \sim t^{-1/3}$, where the interfacial "area" A_{int} is a $d-1$ dimensional hypersurface and L_{sys} is the system size. For interface width w, $A_{int}(t)w/L_{sys}^d = w\epsilon/\sigma$ represents the fraction of the system in the interfacial region. Multiplying the averaged error by $\sigma/(w\epsilon)$ then gives the typical error in the interfacial region. The surface tension corresponding to Eq. (7) is $\sigma = 2\sqrt{2}/3$. We take $w = 2\sqrt{2}$ as a typical measure for the interface width.
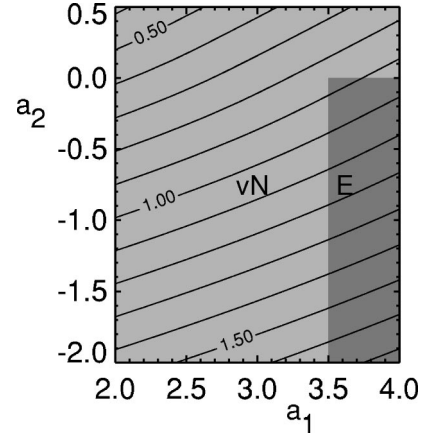

FIG. 4. Contour of scaled error for a single direct update in a 1024^2 system. The systems are evolved well into the scaling regime ($t \approx 3000$) with a fixed-step Euler update. The errors are found by comparing a single direct time step $\Delta t = At^{1/3}$ with two steps of size $\Delta t/2$, and are then scaled by $2\sigma/(A^2 w\epsilon)$ to estimate the average error magnitude per lattice site per A^2 in the interfacial region, as described in the text.

We have investigated this error for a variety of direct algorithms in Fig. 4, where we have plotted the interfacial error as determined above divided by A^2. We plot this error amplitude against a_1 and a_2 for the same shaded regions ["vN" and "E"] as identified in Fig. 1. The typical interfacial error for a given direct step of size $\Delta t = At^{1/3}$ may be obtained by multiplying the appropriate contour value by A^2.

To illustrate the advantages of stable algorithms, as well as of a detailed error analysis where it is possible, we show in Fig. 5 how the error evolves in time for direct steps with $\Delta t = At^{1/3}$ versus the Euler step with fixed Δt. The field ϕ is evolved by the Euler method, and during the evolution error checking is done with single steps that do not contribute to the evolution. The decay of the Euler error shows that the Euler method is asymptotically wastefully accurate.
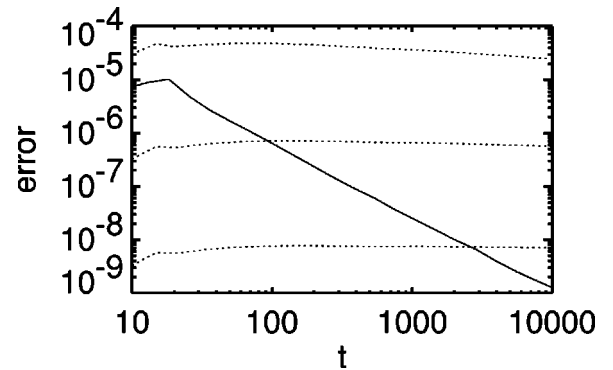

FIG. 5. Plot of scaled error per lattice site near the interface for a single Euler step (solid), and for a single direct step with $a_1 = 3$ and $a_2 = 0$ (dotted with $\Delta t = At^{1/3}$ where from bottom to top $A = 10^{-4}$, 10^{-3}, and 10^{-2}). The scaling of the errors is the same as in Fig. 4, except that the errors are not divided by A^2. For the two smallest A the scaling with A^2 is clearly seen, and so is the time independence of the error for the driven direct step at later times. The system size is 2048^2 and is evolved with a Euler step with $\Delta t = 0.05$.
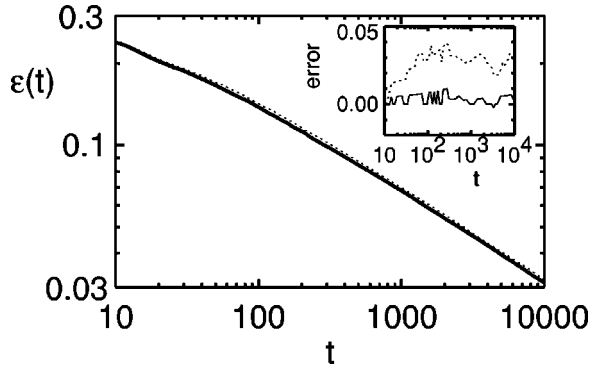
FIG. 6. Plot of ϵ vs t for a Euler update (with $\Delta t = 0.05$, thick solid line) and with the evolution via a direct algorithm ($a_1 = 3$ and $a_2 = 0$) driven with $\Delta t = At^{1/3}$ with $A = 0.1$ (dotted line) and 0.01 (thin solid line) in a 2048^2 system. Until $t = 10$ all systems were evolved with the Euler update. In the inset is plotted the percentage difference between the Euler and direct updates: some error is introduced in the direct steps after $t = 10$ but at later times no increasing deviation from the Euler evolution is seen.

Our single-step analysis and testing does not conclusively demonstrate that an algorithm will be reasonably behaved under successive steps, i.e., there is a possibility of accumulation of error. In Fig. 6 we show the free-energy density for systems evolved by a direct step and compare the evolution to that obtained by the Euler method. It appears that the errors do not accumulate and the free energy decays properly as $t^{-1/3}$.

C. Toward $p > 2$

To go beyond the $p = 2$ steps with $\Delta t \sim t^{1/3}$, it is necessary to find a stable step that satisfies Eq. (37). Comparing with the stability conditions, Eqs. (29), we find only marginally stable algorithms with $a_1 = a_2 = 1/2$ and $a_3 = (3/2 - m)/(3 - m)$ for $0 \leq m \leq 3$. For $m = 0$ this becomes the Crank-Nicholson method, which as noted before, has a fixed time step due to solvability considerations. However, a marginal linear iterative step is possible with $m = 2$ and $a_3 = -1/2$. Unfortunately, whether or not the marginality is a problem, the iterative method (given by Eyre in Ref. [4]) fails to converge absolutely for these parameters. Evidently, then, it is not possible to construct a useful $p = 3$ step from the general step, Eq. (10).

One possible way to develop a $p = 3$ step is to use a method that is both stable and second-order accurate in time. For example, a two-step method that uses both $\phi_{t - \Delta t}$ and ϕ_t to determine the updated field $\phi_{t + \Delta t}$ can be made to have no $O(\Delta t^2)$ error. A preliminary study of vN stability for these two-step methods indicates that these are a possibility.

It is worth considering the prospect of obtaining a $p \to \infty$ step: according to the p classification analysis this would allow the natural $\Delta t \sim t^{2/3}$ time step. However, the error terms need to be strictly proportional to $\partial_t^n \phi$ at each order Δt^n. To achieve this with a one-step method one needs

$$\tilde{\phi}_{t + \Delta t} - (1 - a) \nabla^2 \mu_{t + \Delta t} = \phi_t + a \nabla^2 \mu_t . \tag{40}$$

Equation (30) shows that this step will be linearly unstable when $a > 1/2$ (for large enough Δt), while for $a < 1/2$ one runs into solvability problems. At this point it seems unlikely that a $p \to \infty$ algorithm for the CH equation will be possible.

D. Asymptotic scaling of field derivatives

In this section we derive the relations (33) and (34) that provided the basis for p classification. Enough is known about CH dynamics such that we can explicitly analyze the leading asymptotic decay of mixed space and time derivatives to arbitrary order. We follow the review by Bray [1], and we restrict ourselves to the power-law scaling of these terms at sufficiently late times, where all observable length scales that describe the domain wall morphology, such as the interface curvature radii, are proportional to the domain size $L \sim t^{1/3}$. The domain wall thickness w does not grow with time, so $w \ll L$ asymptotically. However, when analyzing the fields in the interfacial region, defined as the locus of points within a distance w of a domain wall center (i.e., the surface $\phi = 0$), both length scales L and w can appear. The remainder of the system is referred to as the bulk.

The scale of the chemical potential μ is proportional to interface curvature κ due to the Gibbs-Thompson effect, and since $\kappa \sim 1/L$,

$$\mu \sim 1/L \sim 1/t^{1/3} . \tag{41}$$

In the bulk, the chemical potential varies smoothly and continuously, so a Laplacian simply brings in more powers of L:

$$\nabla^2 \mu \sim 1/L^3 \sim 1/t , \tag{42}$$

which implies $\partial_t \phi \sim 1/t$ via the equation of motion (5). Now we use the relation $\phi - \phi_{eq} \sim \mu$ in the bulk [1] to relate derivatives of ϕ and μ. For example, $\nabla^2 \phi \sim \nabla^2 \mu$, so $\partial_t \phi \sim \nabla^2 \phi$. Taking more time derivatives gives

$$\partial_t^n \phi \sim \nabla^2 \partial_t^{n-1} \phi \sim t^{-2/3} \partial_t^{n-1} \phi . \tag{43}$$

Iterating this from the initial value for $\partial_t \phi$ gives $\partial_t^n \phi \sim t^{-(1/3) - (2/3)n}$, the first term in Eq. (34).

When the time derivatives act on a power of the field ϕ^j, the resulting expression contains the j fields and n time derivatives in various combinations. In this case the asymptotic decay comes from the single term proportional to $\phi^{j-1} \partial_t^n \phi$, which means the decay for ϕ^j derivatives is the same as the $j = 1$ case, since the field ϕ is of order unity in the bulk. To illustrate, consider $\partial_t^2 \phi^3 = 6 \phi (\partial_t \phi)^2 + 3 \phi^2 \partial_t^2 \phi$. The second term decays as $t^{-5/3}$ as advertised, while the first term goes as $(t^{-1})^2$ and is asymptotically negligible.

Adding spatial derivatives in the bulk simply brings more factors of L^{-1}, so

$$(\nabla^2)^k \partial_t^{n-1} \phi^j \sim L^{-2k} \partial_t^{n-1} \phi^j \sim t^{-2k/3} t^{-(1/3) - (2/3)n} \tag{44}$$

which gives the second term in Eq. (34).

Near interfaces, ϕ changes by an amount $\Delta \phi_{eq}$ in the amount of time $\tau = w/v \sim t^{2/3}$, it takes an interface to pass by. Therefore we get $\partial_t \phi \sim t^{-2/3}$ in the interfacial region, in con-

trast to $\partial_t \phi \sim t^{-1}$ in the bulk. To determine the scaling $\partial_t^2 \phi$, consider sitting at a point just outside the interfacial region, in front of the moving interface. At a time $O(\tau)$ later this point will be in the interfacial region, so $\partial_t \phi$ will have changed from a bulk to an interfacial value. This gives

$$\partial_t^2 \phi \sim (t^{-2/3} - t^{-1})/\tau \sim t^{-4/3}. \tag{45}$$

Repeating this argument for higher derivatives gives $\partial_t^n \phi \sim t^{-2n/3}$ in the interface, the first term in Eq. (33).

For time derivatives of ϕ^j at the interface, we again get multiple terms with the various combinations of n time derivatives and j fields. In this case, however, every term contributes to the asymptotic decay. Essentially every time derivative, wherever it acts, brings a factor of $t^{-2/3}$, and these are the only factors causing the decay. Hence $\partial_t^n \phi^j \sim \partial_t^n \phi$. Finally, adding spatial derivatives in the interfacial region brings factors of w^{-1} rather than L^{-1}, and so does not change the asymptotic decay. This proves the second relation in Eq. (33).

IV. COMPUTATIONAL ADVANTAGE

Having established the possibility of controlled accuracy CH simulation with a growing step size $\Delta t \sim t^\alpha$, we now explore the relative computational advantage offered by such an algorithm. As described in Sec. I, the goal in such simulations is to evolve as close as possible to the scaling regime, meaning the largest possible $L(t)$. This means evolving until finite-size effects enter, since stopping earlier means a smaller system size could be chosen. Finite-size effects are expected to appear when $L(t) \sim L_0 t^{1/3}$ is some fraction of the system size, so we define the simulation ending time t_{max} by $L(t_{max}) = f L_{sys}$ or

$$t_{max} = (f L_{sys}/L_0)^3 = (f \Delta x N/L_0)^3, \tag{46}$$

where N is the linear size of the lattice and f is a small constant factor. There is some arbitrariness in the definition of the length scale $L(t)$ that simply factors into the optimal choice for f. We take the inverse interface density as our measure, that is,

$$L(t) = \frac{L_{sys}^d}{A_{int}} = \frac{\sigma}{\epsilon(t)} = \frac{\sigma}{\epsilon_0} t^{1/3} \tag{47}$$

using the interfacial area A_{int} from Sec. III B, and its relation to the free-energy density and surface tension derived therein. From our data in $d=2$ we find $\epsilon_0 \simeq 0.675$, so we take $L_0 = \sigma/\epsilon_0 \simeq 1.40$.

Evolving to t_{max} with the Euler step (or any fixed-size step) requires $n = t_{max}/\Delta t_0$ steps, where Δt_0 is the step size. For our square lattice with $\Delta x = 1$ we find $\Delta t_0 = 0.05$ is close to the maximum stable value. More generally, one expects $\Delta t_0 \sim \Delta x^4$ [2]. Evolving to a late time t_{max} with a growing step size $\Delta t \sim A t^\alpha \sim dt/dn$ requires a number of steps

$$n \sim \int_{t_0}^{t_{max}} A^{-1} t^{-\alpha} dt \sim \frac{1}{A(1-\alpha)} t_{max}^{1-\alpha}, \tag{48}$$

where a fixed-size step is used until some early time t_0. Since $t_0 \ll t_{max}$, the t_0-dependent terms can be neglected.

Finally, we determine empirically the ratio of computer time per step $\beta = \tau_{stable}/\tau_{Euler}$. For direct steps, the FFT involved implies $\beta \sim \ln N$. For lattices of size 1024^2–4096^2 we find $\beta \simeq 2.3 \pm 0.1$.

Putting all this together, we find the ratio of computer time cost for Euler steps to stable steps to be

$$\frac{A(1-\alpha)t_{max}^\alpha}{\beta \Delta t_0} = \frac{A(1-\alpha)}{\beta \Delta t_0} \left(\frac{f \Delta x}{L_0}\right)^{3\alpha} N^{3\alpha}. \tag{49}$$

For direct steps, $\alpha = 1/3$, so the relative speedup over Euler integration grows with the system size as $N/\ln N$. From $\Delta t_0 \sim \Delta x^4$ we also see the speedup factor scaling as $1/\Delta x^3$, making stable steps an optimal choice when a smaller lattice spacing is desired. A $p=3$ algorithm has $\alpha = 4/9$ and offers a speedup factor of $N^{4/3}/\ln N$.

We conclude by plugging in reasonably conservative parameter values. From Fig. 4 we see that the typical interfacial error for the $a_1 = 3$, $a_2 = 0$ direct step is about $0.7 A^2$. This is to be compared to $\Delta \phi_{eq} = 2$, the range in which ϕ varies. The choice $A = 0.1$ is shown in Fig. 6 to give an error in the free-energy density around 3% of the Euler value. While this seems perhaps high, we note that this is comparable and probably smaller than the error already introduced in the Euler discretization of the continuum CH equation due to the large lattice constant. An interesting question for future study is what choice of Δx and A will give optimal accuracy and efficiency. We conclude that $A = 0.1$ is a reasonable choice. We also take $\alpha = 1/3$, $f = 1/10$, $\beta = 2.5$, $\Delta x = 1$, and L_0 as given above. These combine to give a factor $0.038N$. For a 1024^2 lattice the direct step is a factor 40 faster than the Euler method, while for a 8192^2 lattice it is a factor 300 faster

V. CONCLUSIONS AND FUTURE DIRECTIONS

We have seen that the general Cahn-Hilliard (CH) step, Eq. (10), provides a range of linearly stable algorithms that prove to be gradient stable for enormous single time steps up to $\Delta t = 10^{10}$. With these steps unphysical instabilities arising from the discrete implementations are no longer the limiting factor. Instead, accuracy considerations dominate. For conserved Cahn-Hilliard coarsening, we have analyzed and tested the accuracy scaling for single dynamical time steps that *increase* without bound with time as $\Delta t \sim t^\alpha$. We find that the errors are dominated by terms of order Δt^p where they are no longer proportional to $\partial_t^p \phi$. These dominant errors restrict the growth of the time step to grow as $\Delta t \sim t^{2(p-1)/(3p)}$, which approaches the natural dynamical time step $\tau \sim t^{2/3}$ only as $p \to \infty$. The Euler method, by contrast, is restricted to a constant Δt. This is also the case for existing implicit Fourier spectral algorithms. The direct steps obtained from Eq. (10) with $a_3 = 1$ are linear and diagonalized in Fourier space, and so can be simply integrated via FFT's. This requires no more computer memory than the Euler step, since the FFT can be performed in place. A range of parameters, described by the shaded boxes in Fig. 1, are stable.

These direct steps exhibit $p=2$ and so allow $\Delta t \sim t^{1/3}$, which results in speedup factors proportional to the linear size of the system.

Future work in further developing these methods for the CH equation includes determining possible $p=3$ algorithms, for which $\Delta t \sim t^{4/9}$ is possible and the relative speedup over the Euler method is of order $N^{4/3}/\ln N$. Our preliminary work has shown that $O(\Delta t^2)$ accurate two-step methods can be made unconditionally vN stable. It remains to test these stability predictions numerically to see if useful $p=3$ algorithms are possible.

It is straightforward to construct a Fourier spectral method integration algorithm for the stable steps analyzed here. In fact, the numerical cost of the spectral method would be quite small, since the direct steps already employ FFT's for solving the update equation. The primary benefit of the spectral method for unstable algorithms is that it significantly enhances the maximum Δt_0 allowed by stability. It is not clear how much benefit spectral methods would bring to an already stable algorithm, but this should be explored.

With the Euler step, the simulation efficiency was strongly dependent on Δx, leading to choosing values that were as large as feasible. Consequently the interface profile is typically poorly resolved, modifying and introducing significant anisotropy into the surface tension. In contrast, the efficiency of these stable methods is much less dependent on the choice of lattice size, making them a useful tool in applications where a more accurate interface profile is desired.

Our analysis has been for errors after a single time step. Gradient stability would seem to permit errors to only modify interfacial properties in a curvature- or velocity-dependent way, either of which would be consistent with an effective continuum dynamics that still monotonically decreases the overall energy density from any field configuration. For the CH equation at least, our observed $\epsilon \sim t^{-1/3}$ decay of the free energy, even when $\Delta t \sim A t^{1/3}$, indicates that there is no significant curvature-dependent modification of interfacial speeds. We are studying the relationship between single-step errors and errors of the asymptotic scaling functions describing correlations in more detail.

We feel that our basic approach should be applicable in a wide variety of scalar or vector phase-field systems that have both nonlinearities and numerical instabilities. There are just three basic ingredients: (i) allow for a general semi-implicit parametrization, following Eq. (10); (ii) check for unconditional von Neumann (linear) stability of an individual update step, following Sec. II A; and (iii) numerically test the vN stable algorithms for speed, accuracy, and nonlinear stability in order to pick the best parameters for further study. As long as the stability criteria are lattice independent, the resulting algorithms can be used with any regular lattice in any spatial dimension, and even with irregular discretizations such as used in adaptive mesh techniques.

The detailed accuracy analysis and the p classification scheme of the CH equation does not immediately generalize to other cases, but we stress that this analysis is not essential. Once the stable steps are identified, a variable step size method [9] will tune the time step to its accuracy-limited value. Finally, we note that the inclusion of additive noise

should be compatible with gradient stable algorithms, as long as the noise amplitude is used to extend the range of possible ϕ_0^2 used in the vN analysis.

APPENDIX A: EYRE'S THEOREM

We repeat Eyre's stability theorem [3] here to flesh out the derivation for the conserved dynamics case, and to clarify some details of the proof. In particular, there are a few equations in Ref. [3] that lack factors of vector norms. More substantively, we find that Eyre's theorem as originally presented was slightly more restrictive than necessary.

A central quantity in Eyre's theorem is the Hessian matrix

$$M_{ij} = \frac{\partial^2 F}{\partial \phi_i \partial \phi_j}, \tag{A1}$$

where F is the free energy and ϕ_i represents the field at the lattice site i (we consider only scalar one-component fields here). For free energies of interest in coarsening, this matrix has both positive and negative eigenvalues. Eyre finds a stable first-order step by splitting the free energy into *contractive* and *expansive* parts, $F = F^C + F^E$, such that F^C is convex and F^E is concave; that is, the eigenvalues of M_{ij}^C, the Hessian matrix corresponding to F^C, are strictly nonnegative, and the eigenvalues of M_{ij}^E corresponding to F^E are strictly nonpositive for any possible field configuration.

Let $\lambda_{min} < 0$ represent the lower bound for the eigenvalues of M over all fields $\boldsymbol{\phi}$ (such a bound must exist [3]), and $\lambda_{max}^E \leq 0$ represent the upper bound on the eigenvalues of M^E. The main result is that if

$$\lambda_{max}^E \leq \frac{1}{2} \lambda_{min}, \tag{A2}$$

then the field equations of motion

$$\phi_{t+\Delta t} + \Delta t \left. \frac{\delta F^C}{\delta \phi} \right|_{\phi_{t+\Delta t}} = \phi_t - \Delta t \left. \frac{\delta F^E}{\delta \phi} \right|_{\phi_t} \tag{A3}$$

for nonconserved dynamics or

$$\phi_{t+\Delta t} - \Delta t \nabla^2 \left. \frac{\delta F^C}{\delta \phi} \right|_{\phi_{t+\Delta t}} = \phi_t + \Delta t \nabla^2 \left. \frac{\delta F^E}{\delta \phi} \right|_{\phi_t} \tag{A4}$$

for conserved dynamics lead to a strict nonincrease of the free energy in time:

$$F(\boldsymbol{\phi}_{t+\Delta t}) \leq F(\boldsymbol{\phi}_t), \tag{A5}$$

where we have suppressed the lattice index for clarity. This holds unconditionally for all field configurations $\boldsymbol{\phi}_t$ and all step sizes $\Delta t > 0$. Convexity of F^C ensures that the implicit equation for $\boldsymbol{\phi}_{t+\Delta t}$ has a unique solution.

The energy dissipation property, along with other reasonable requirements such as positivity of F, is called *gradient stability* by Eyre [3]. While gradient stability can be obtained for many algorithms, such as the Euler step, by using a small enough Δt, the algorithm defined by Eqs. (A2)–(A4) guarantees it for *arbitrarily large* Δt. Even so, finding the splittings into F^C and F^E that lead to Eq. (A2) can be a difficult task, and the splittings, if they exist, may not be unique.

Condition (A2) corrects the corresponding condition in Ref. [3], $\lambda_{max}^E \leq \lambda_{min}$. The current form is less restrictive since $\lambda_{min} < 0$.

An extremely useful corollary to Eyre's theorem is that if the eigenvalue condition (A2) is satisfied for a restricted set of fields $\boldsymbol{\phi}$, then Eq. (A5) still applies for all Δt, provided $\boldsymbol{\phi}_t$ always stays within this restricted set. For example, $\boldsymbol{\phi}$ could be field configurations with $\phi_i^2 < \phi_0^2$ for all i, for some constant ϕ_0. This can be useful when $\boldsymbol{\phi}$ is physically restricted by the dynamics, and is employed in the direct algorithms discussed in Sec. II A

The proof of Eq. (A5) relies on two inequalities

$$F(\boldsymbol{\phi}_{t+\Delta t}) - F(\boldsymbol{\phi}_t) \leq \sum_i \left. \delta\phi_i \frac{\partial F}{\partial \phi_i}\right|_{\boldsymbol{\phi}_{t+\Delta t}} - \frac{1}{2}\lambda_{min}|\delta\boldsymbol{\phi}|^2 \tag{A6}$$

and

$$\sum_i \delta\phi_i \left(\left.\frac{\partial F^E}{\partial \phi_i}\right|_{\boldsymbol{\phi}_{t+\Delta t}} - \left.\frac{\partial F^E}{\partial \phi_i}\right|_{\boldsymbol{\phi}_t} \right) \leq \lambda_{max}^E |\delta\boldsymbol{\phi}|^2, \tag{A7}$$

where $\delta\phi_i \equiv \phi_{i,t+\Delta t} - \phi_{i,t}$ and $|\delta\boldsymbol{\phi}|^2 = \Sigma_i \delta\phi_i^2$. These are simply properties of multivariable functions, and are derived in appendix B for completeness.

Consider first *nonconserved dynamics*. By adding $\Delta t[\partial F^E/\partial \phi_i]_{\boldsymbol{\phi}_{t+\Delta t}}$ to both sides of the equation of motion, Eq. (A3), one obtains

$$\left.\frac{\partial F}{\partial \phi_i}\right|_{\boldsymbol{\phi}_{t+\Delta t}} = -\frac{1}{\Delta t}\delta\phi_i + \left.\frac{\partial F^E}{\partial \phi_i}\right|_{\boldsymbol{\phi}_{t+\Delta t}} - \left.\frac{\partial F^E}{\partial \phi_i}\right|_{\boldsymbol{\phi}_t}. \tag{A8}$$

Substituting this into Eq. (A6) gives

$$F(\boldsymbol{\phi}_{t+\Delta t}) - F(\boldsymbol{\phi}_t) \leq \sum_i \delta\phi_i \left(\left.\frac{\partial F^E}{\partial \phi_i}\right|_{\boldsymbol{\phi}_{t+\Delta t}} - \left.\frac{\partial F^E}{\partial \phi_i}\right|_{\boldsymbol{\phi}_t} \right)$$
$$- \left(\frac{1}{2}\lambda_{min} + \frac{1}{\Delta t} \right)|\delta\boldsymbol{\phi}|^2. \tag{A9}$$

Next, use Eq. (A7) to complete the proof:

$$F(\boldsymbol{\phi}_{t+\Delta t}) - F(\boldsymbol{\phi}_t) \leq \left(\lambda_{max}^E - \frac{1}{2}\lambda_{min} - \frac{1}{\Delta t} \right)|\delta\boldsymbol{\phi}|^2,$$
$$\leq 0, \tag{A10}$$

where the last inequality follows by assumption Eq. (A2).

Analyzing *conserved dynamics* is complicated by the Laplacian in the equations of motion. Consider a general dimensional lattice of n sites with lattice Laplacian $(\nabla^2)_{ij} \equiv A_{ij}$ a symmetric $n \times n$ matrix with eigenvalues $\lambda_1 = 0$ and $\lambda_m < 0$ for all $m > 1$. Let $u_i^{(m)}$ represent the ith component of the mth eigenvector of A, then we can write the Kronecker δ function as

$$\delta_{ik} = \sum_{m=1}^n u_i^{(m)} u_k^{(m)} = \sum_{j=1}^n \tilde{A}_{ij} A_{jk} + u_i^{(1)} u_k^{(1)}, \tag{A11}$$

where the pseudoinverse $\tilde{A}$ is defined by

$$\tilde{A}_{ij} = \sum_{m \neq 1}^n \frac{1}{\lambda_m} u_i^{(m)} u_j^{(m)}. \tag{A12}$$

Note that the eigenvalue $\lambda_1 = 0$ corresponds to the eigenvector $u_i^{(1)} = 1/\sqrt{n}$ for all i, i.e., a uniform field. Now we insert Eq. (A11) into the sum in Eq. (A6) and sum on k to get

$$F(\boldsymbol{\phi}_{t+\Delta t}) - F(\boldsymbol{\phi}_t) \leq \sum_{i,j,k} \left. \delta\phi_i \tilde{A}_{ij} A_{jk} \frac{\partial F}{\partial \phi_k}\right|_{\boldsymbol{\phi}_{t+\Delta t}}$$
$$- \frac{1}{2}\lambda_{min}|\delta\boldsymbol{\phi}|^2, \tag{A13}$$

where we have used $\Sigma_i \delta\phi_i = 0$, which follows from the conservation law. Proceeding by analogy with the nonconserved case, we subtract $\Delta t A_{jk}[\partial F^E/\partial \phi_k]_{\boldsymbol{\phi}_{t+\Delta t}}$ from both sides of the equation of motion, Eq. (A4), to get

$$\sum_k \left. A_{jk} \frac{\partial F}{\partial \phi_k}\right|_{\boldsymbol{\phi}_{t+\Delta t}} = \frac{\delta\phi_j}{\Delta t} + \sum_k A_{jk} \left(\left.\frac{\partial F^E}{\partial \phi_k}\right|_{\boldsymbol{\phi}_{t+\Delta t}} - \left.\frac{\partial F^E}{\partial \phi_k}\right|_{\boldsymbol{\phi}_t} \right). \tag{A14}$$

Substituting this into Eq. (A13) gives

$$F(\boldsymbol{\phi}_{t+\Delta t}) - F(\boldsymbol{\phi}_t) \leq \sum_i \delta\phi_i \left(\left.\frac{\partial F^E}{\partial \phi_i}\right|_{\boldsymbol{\phi}_{t+\Delta t}} - \left.\frac{\partial F^E}{\partial \phi_i}\right|_{\boldsymbol{\phi}_t} \right)$$
$$- \frac{1}{2}\lambda_{min}|\delta\boldsymbol{\phi}|^2 + \frac{1}{\Delta t} \sum_{i,j} \delta\phi_i \delta\phi_j \tilde{A}_{ij}, \tag{A15}$$

which is identical to Eq. (A9) except for the $1/\Delta t$ term. From the definition of $\tilde{A}$ and an expansion of $\delta\boldsymbol{\phi}$ in the eigenvalues $\mathbf{u}^{(m)}$ it follows that

$$\sum_{i,j} \delta\phi_i \delta\phi_j \tilde{A}_{ij} \leq 0, \tag{A16}$$

so this term can be dropped from the right-hand side of Eq. (A15), and the proof follows as before to yield Eq. (A5).

APPENDIX B: INEQUALITIES USED IN EYRE'S THEOREM

For completeness, we rederive Eqs. (A6) and (A7) here. Consider a general function $f(\mathbf{x})$ of n variables $\mathbf{x} = (x_1, \ldots, x_n)$. From the fundamental theorem of calculus

$$f(\mathbf{x}+\mathbf{y}) - f(\mathbf{x}) = \sum_i y_i \int_0^1 ds_1 \frac{\partial f}{\partial x_i}\bigg|_{\mathbf{x}+s_1\mathbf{y}}, \qquad (B1)$$

that is, we introduce the parameter s_1 to integrate along the "diagonal" path from $\mathbf{x}$ to $\mathbf{x}+\mathbf{y}$. Similarly, we can write

$$\frac{\partial f}{\partial x_i}\big|_{\mathbf{x}+s_1\mathbf{y}} - \frac{\partial f}{\partial x_i}\bigg|_{\mathbf{x}} = \sum_j y_j \int_0^{s_1} ds_2 \frac{\partial^2 f}{\partial x_i \partial x_j}\bigg|_{\mathbf{x}+s_2\mathbf{y}}. \qquad (B2)$$

Combining these gives the identity

$$f(\mathbf{x}+\mathbf{y}) - f(\mathbf{x})$$
$$= \sum_i y_i \frac{\partial f}{\partial x_i}\bigg|_{\mathbf{x}} + \int_0^1 ds_1 \int_0^{s_1} ds_2 \sum_{i,j} y_i y_j \frac{\partial^2 f}{\partial x_i \partial x_j}\bigg|_{\mathbf{x}+s_2\mathbf{y}}. \qquad (B3)$$

Now consider the case where the eigenvalues of the matrix $M_{ij} = \partial^2 f/\partial x_i \partial x_j$ are bounded from below by some constant λ_{min} for all $\mathbf{x}$. In this case

$$\sum_{i,j} y_i y_j \frac{\partial^2 f}{\partial x_i \partial x_j}\bigg|_{\mathbf{x}+s_2\mathbf{y}} \geq \lambda_{min}|\mathbf{y}|^2, \qquad (B4)$$

which follows straightforwardly from an expansion of $\mathbf{y}$ in the basis of eigenvectors of M, with $|\mathbf{y}|^2 = \Sigma_i y_i^2$. Thus we have

$$f(\mathbf{x}+\mathbf{y}) - f(\mathbf{x}) \geq \sum_i y_i \frac{\partial f}{\partial x_i}\bigg|_{\mathbf{x}} + \frac{1}{2}\lambda_{min}|\mathbf{y}|^2, \qquad (B5)$$

where the 1/2 follows from the s integrals. Taking the function f to be the free energy F with $\mathbf{x} = \boldsymbol{\phi}_{t+\Delta t}$ and $\mathbf{y} = \boldsymbol{\phi}_t - \boldsymbol{\phi}_{t+\Delta t}$ results in Eq. (A6).

The second inequality results from setting $s_1 = 1$ in Eq. (B2), then multiplying by y_i, and summing

$$\sum_i y_i \left(\frac{\partial f}{\partial x_i}\bigg|_{\mathbf{x}+\mathbf{y}} - \frac{\partial f}{\partial x_i}\bigg|_{\mathbf{x}}\right) = \sum_{i,j} y_i y_j \int_0^1 ds \frac{\partial^2 f}{\partial x_i \partial x_j}\bigg|_{\mathbf{x}+s\mathbf{y}}. \qquad (B6)$$

We then use a relation similar to Eq. (B4), only with the eigenvalues of $\partial^2 f/\partial x_i \partial x_j$ assumed to be bounded *above* by λ_{max}, to get

$$\sum_i y_i \left(\frac{\partial f}{\partial x_i}\bigg|_{\mathbf{x}+\mathbf{y}} - \frac{\partial f}{\partial x_i}\bigg|_{\mathbf{x}}\right) \leq \lambda_{max}|\mathbf{y}|^2. \qquad (B7)$$

Now we can take $f = F^E$ and $\mathbf{x}$ and $\mathbf{y}$ as before to get Eq. (A7).

---

[1] A.J. Bray, Adv. Phys. **43**, 357 (1994).

[2] T.M. Rogers, K.R. Elder, and R.C. Desai, Phys. Rev. B **37**, 9638 (1988).

[3] D. J. Eyre, http://www.math.utah.edu/~eyre/research/methods/stable.ps

[4] D. J. Eyre, in *Computational and Mathematical Models of Microstructural Evolution*, edited by J. W. Bullard *et al.* (Materials Research Society, Warrendale, PA, 1998), pp. 39–46.

[5] N. Provatas, N. Goldenfeld, and J. Dantzig, Phys. Rev. Lett. **80**, 3308 (1998).

[6] Y. Oono and S. Puri, Phys. Rev. A **38**, 434 (1988).

[7] L.-Q. Chen and J. Shen, Comput. Phys. Commun. **108**, 147 (1998).

[8] J. Zhu, L.-Q. Chen, J. Shen, and V. Tikare, Phys. Rev. E **60**, 3564 (1999).

[9] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical Recipes in C*, 2nd ed. (Cambridge University Press, New York, 1993).

[10] W. F. Ames, *Numerical Methods for Partial Differential Equations*, 3rd ed. (Academic Press, San Diego, 1992).

[11] For regions not near an interface, the field is given by $\phi - \phi_{eq} \propto \mu$. For the interior regions of curved interfaces, μ and ϕ_{eq} have the same sign, and thus result in a value $|\phi| > |\phi_{eq}|$. Note that this excess is of order $t^{-1/3}$.